

.NET Framework And Tools

**Anders Hejlsberg
Distinguished Engineer**

Microsoft®
PDC 2000
Professional Developers Conference

Microsoft®
.net

the defining

point

.NET Blueprint

Visual
Studio.
NET

Your Application
and Web Service

Your Internal
Services

Orchestration

.NET
Framework

Operations

Windows
CE, ME,
2000, .NET

Internet Protocols
SOAP "blue book"
HTTP, SMTP, XML

.NET Enterprise
Servers

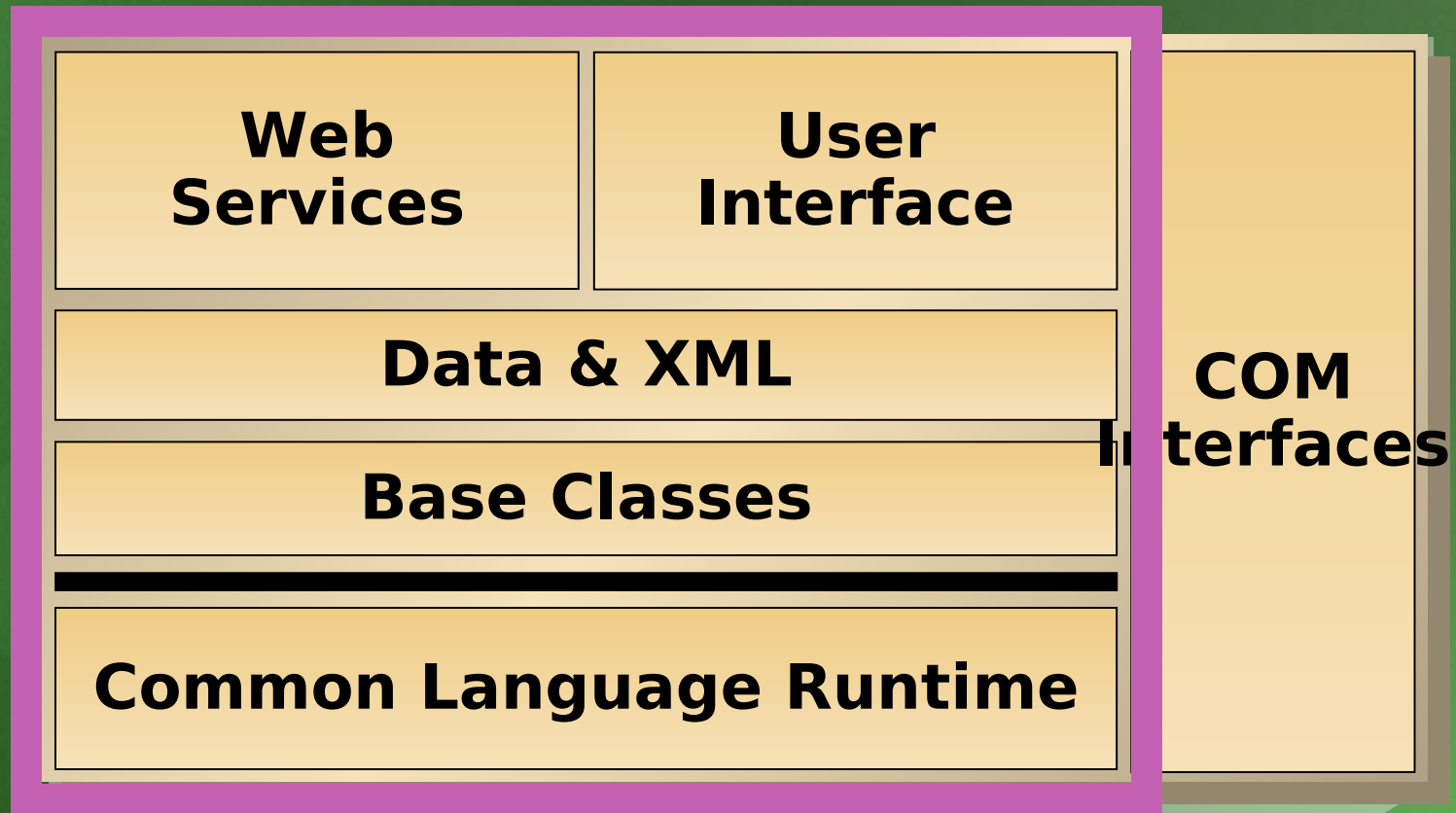
Building Block
Services

3rd Party Web Service

Applications
Using Your
Service

End-User
Clients

.NET Framework



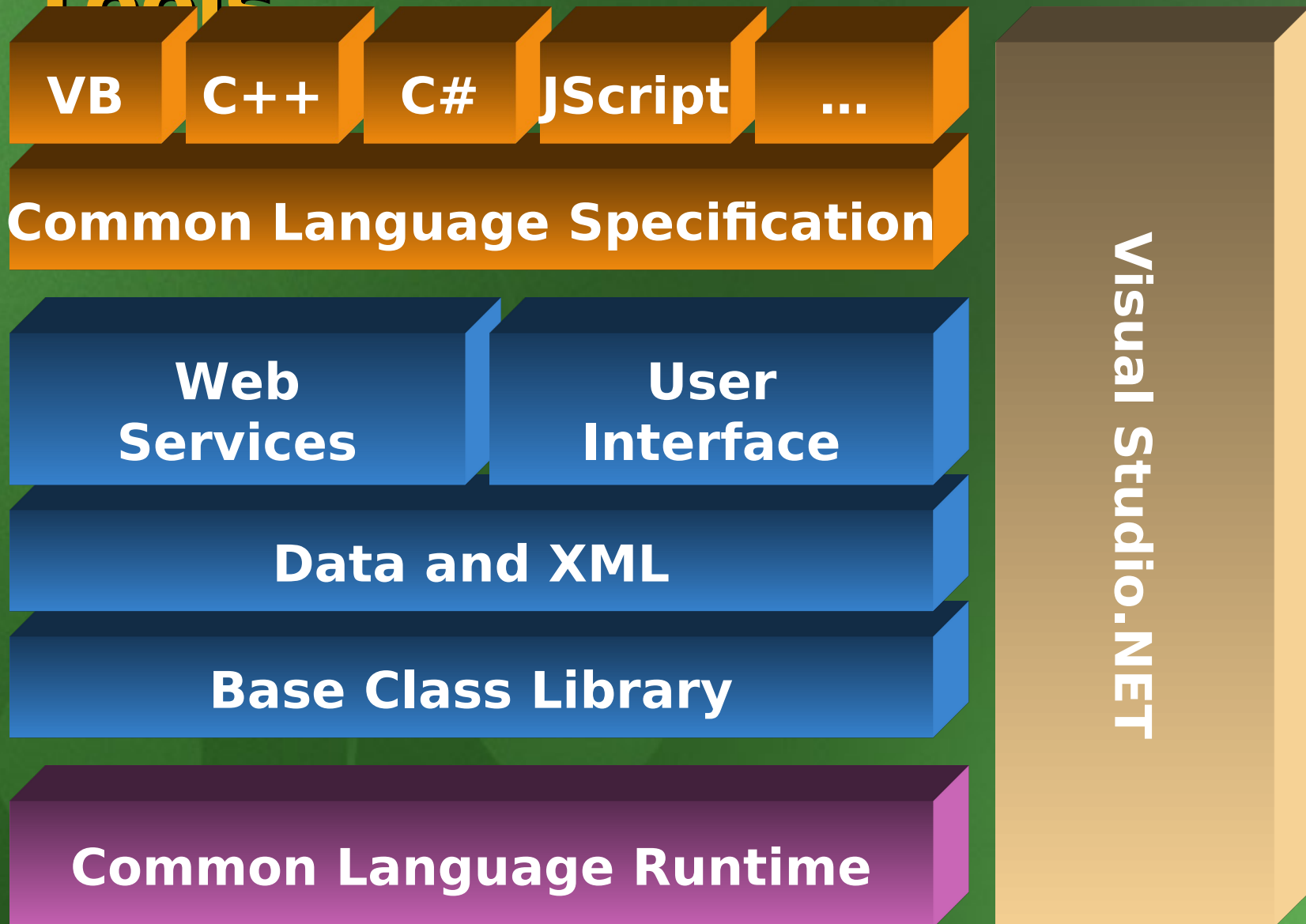
Microsoft®
PDC 2000
Professional Developers Conference

Microsoft®
.net

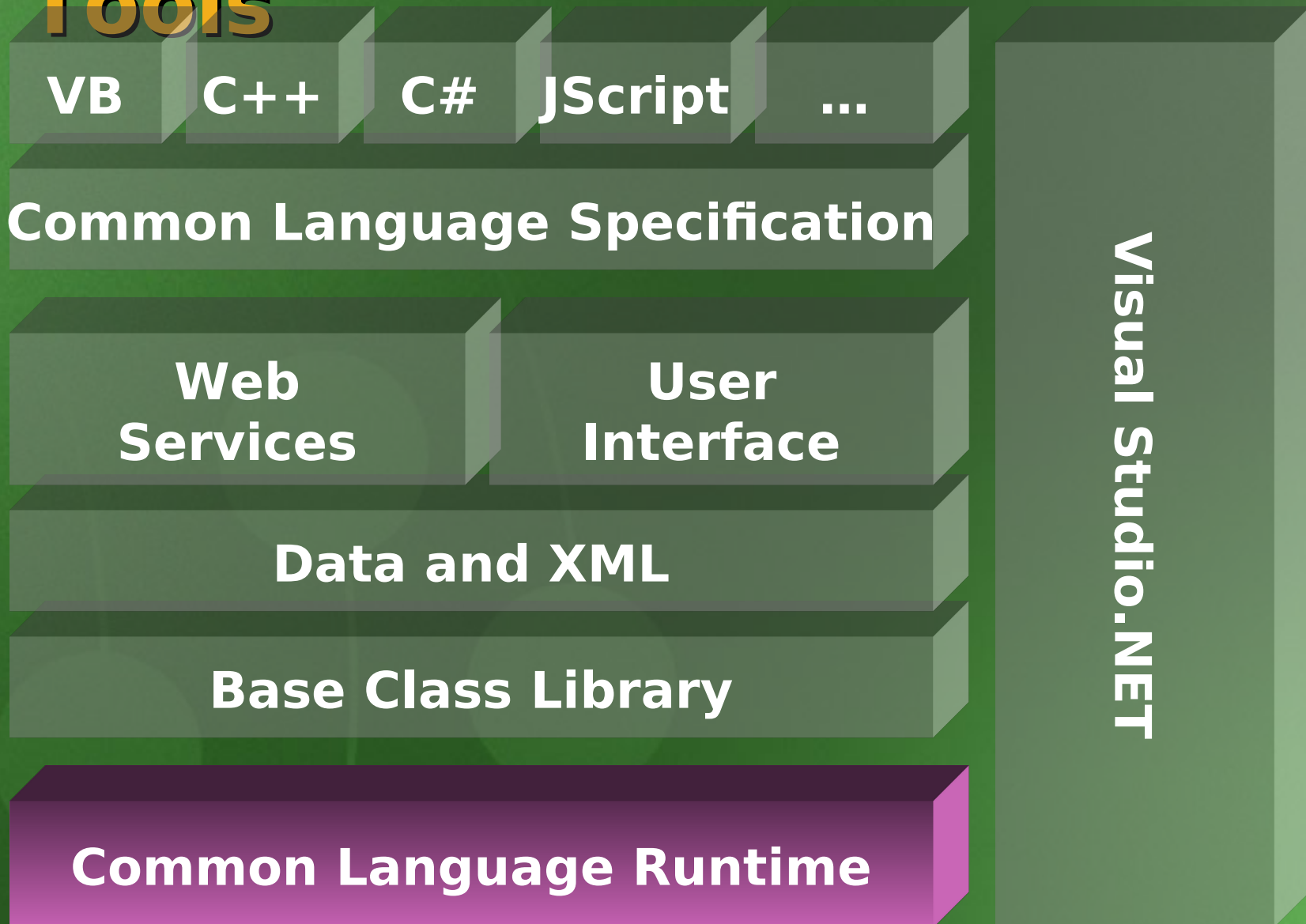
the defining

point

Framework, Languages, And Tools



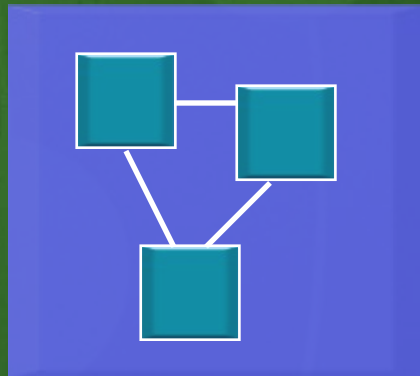
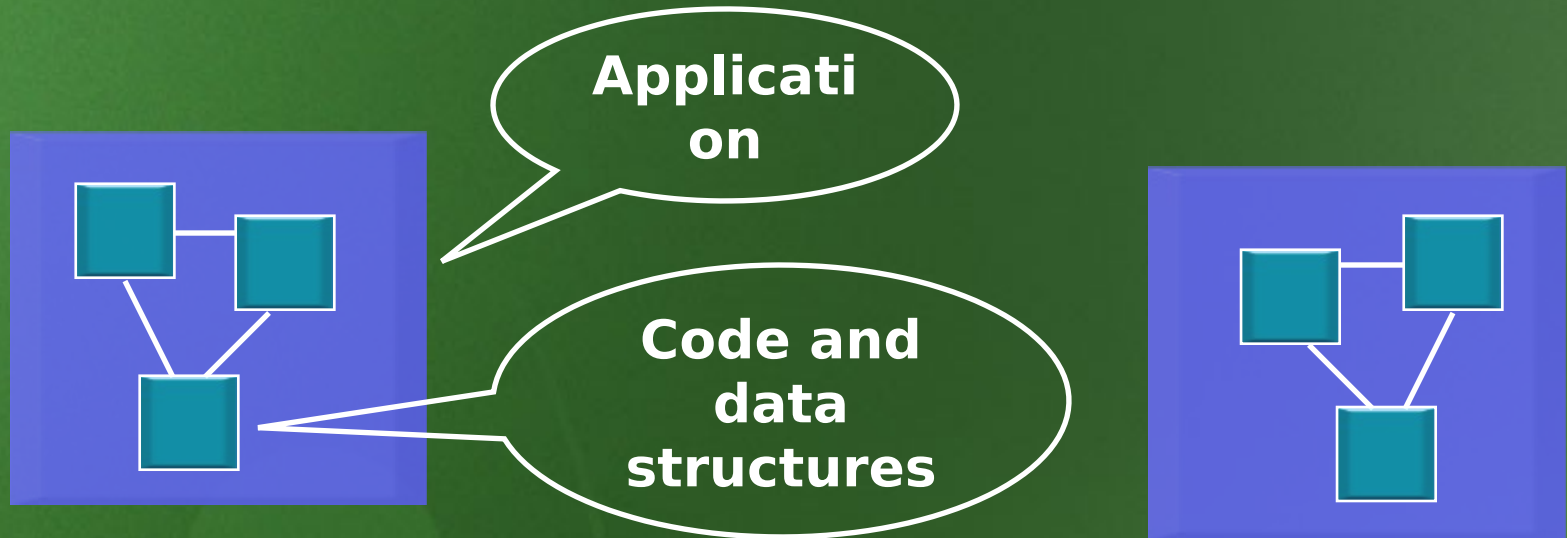
Framework, Languages, And Tools



Common Language Runtime Design Goals

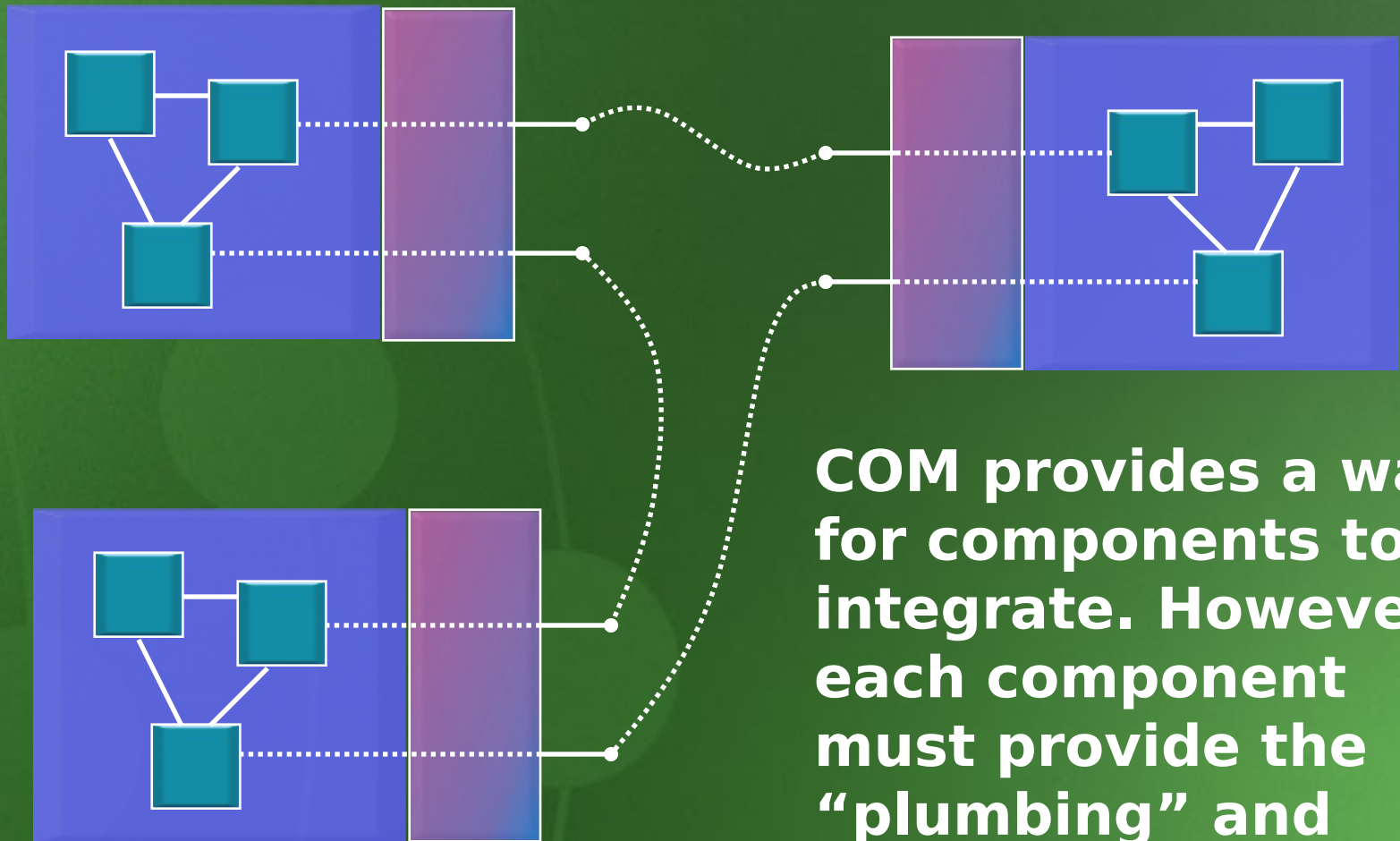
- **Dramatically simplify application development**
- **Provide a robust and secure execution environment**
- **Support multiple programming languages**
- **Simplify deployment and management**

The .NET Evolution



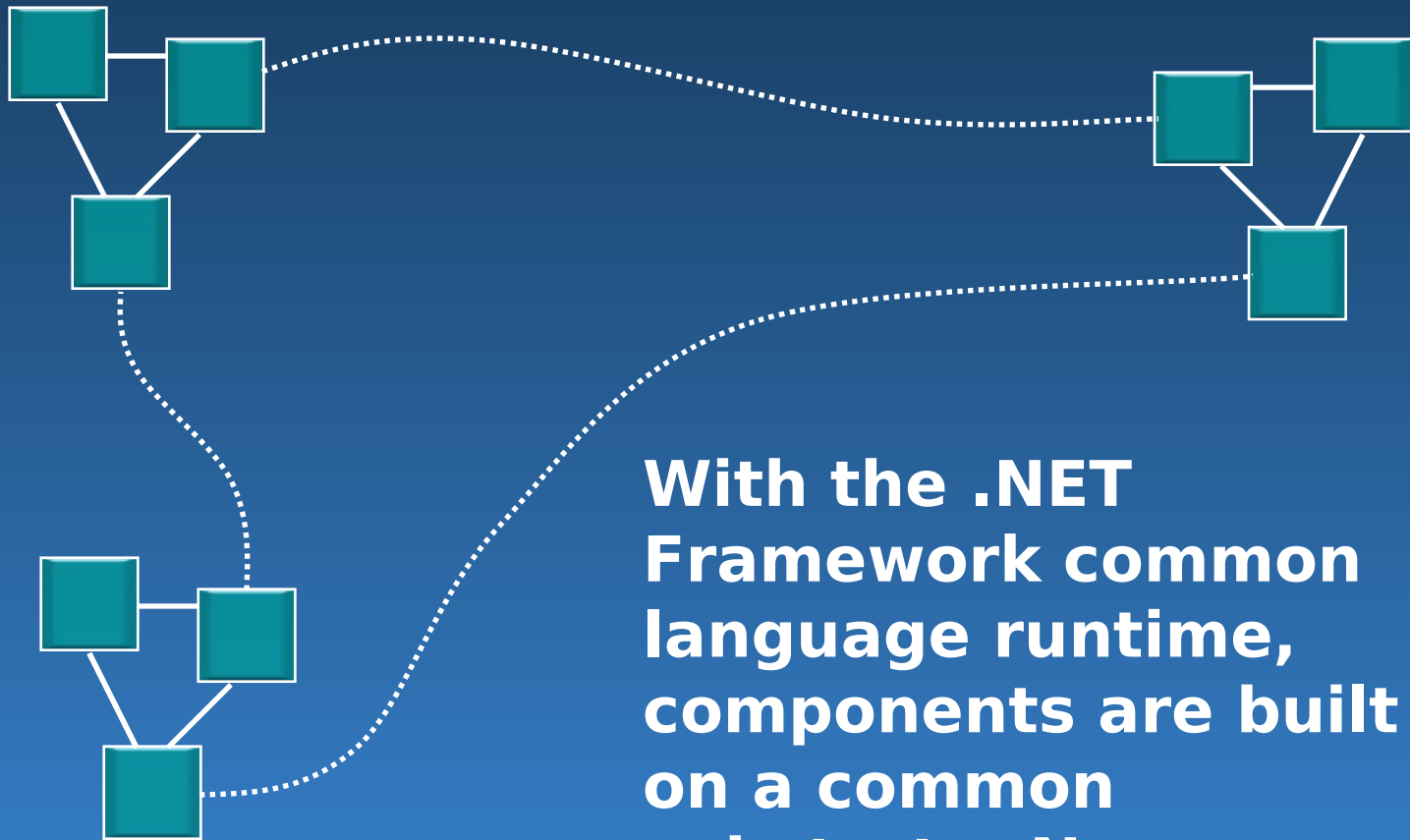
Before COM, applications were completely separate entities with little or no integration

The .NET Evolution



COM provides a way for components to integrate. However, each component must provide the “plumbing” and objects cannot directly interact.

The .NET Evolution



With the .NET Framework common language runtime, components are built on a common substrate. No “plumbing” is needed and objects can directly

Simplify Development

- **Completely eliminates plumbing**
 - No registration, GUIDs, .IDL files, HRESULTs, IUnknown, AddRef/Release, CoCreateInstance, etc.
- **Object Oriented to the core**
 - Classes and inheritance fully supported
 - Even across languages!
- **Seamless integration**
 - Any .NET class can be used as a COM class with zero extra work
 - COM classes can be imported as .NET classes

Common Language

Base Class Library Support

Thread Support

COM Marshaler

Type Checker

Exception Manager

Security Engine

Debug Engine

**IL to Native
Compilers**

**Code
Manager**

**Garbage
Collector**

Class Loader

Robust And Secure

- **Automatic lifetime management**
 - All .NET objects are garbage collected
 - No stray pointers, no circular references
 - Multi-generational mark-and-compact GC
 - Self-configuring, dynamically tuning
- **Exception handling**
 - Error handling is a 1st class concept (not bool or HRESULTs)

Robust And Secure

- **Several compilation models**
 - Native (e.g. Managed C++)
 - MSIL (e.g. VB and C#)
 - No interpreter: Install-time or run-time IL to native compilation
- **Code correctness and type-safety**
 - IL can be verified to guarantee type-safety
 - No unsafe casts, no uninitialized variables, no out-of-bounds array indexing
- **Evidence-based security**
 - Based on origin of code as well as user
 - Extensible permissions

Multi-Language Platform

- The freedom to choose language
 - All features of .NET platform available to any .NET programming language
 - Application components can be written in multiple languages
- Highly leveraged tools
 - Debuggers, profilers, code coverage analyzers, etc. work for all languages

Simplify Deployment And Management

Assemblies

- The unit of deployment, versioning, and security
- Like DLLs, but self-describing through manifest

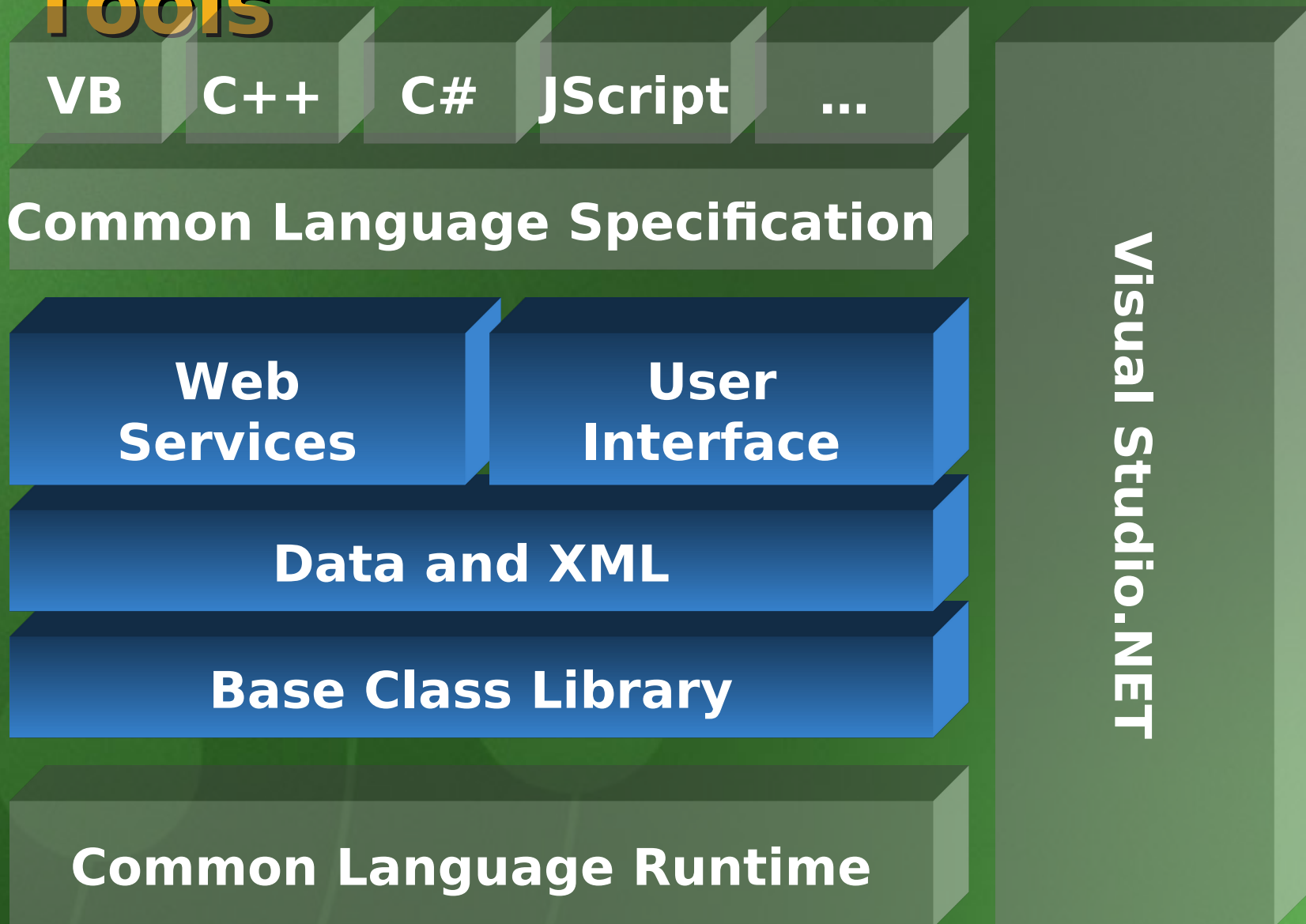
■ Zero-impact install

- Applications and components can be shared or private

■ Side-by-side execution

- Multiple versions of the same component can co-exist, even in the same process

Framework, Languages, And Tools



Framework Design

Goals

- Web standards and practices as the foundation
- Unify application models
- Make it simple to use
- Factored and extensible

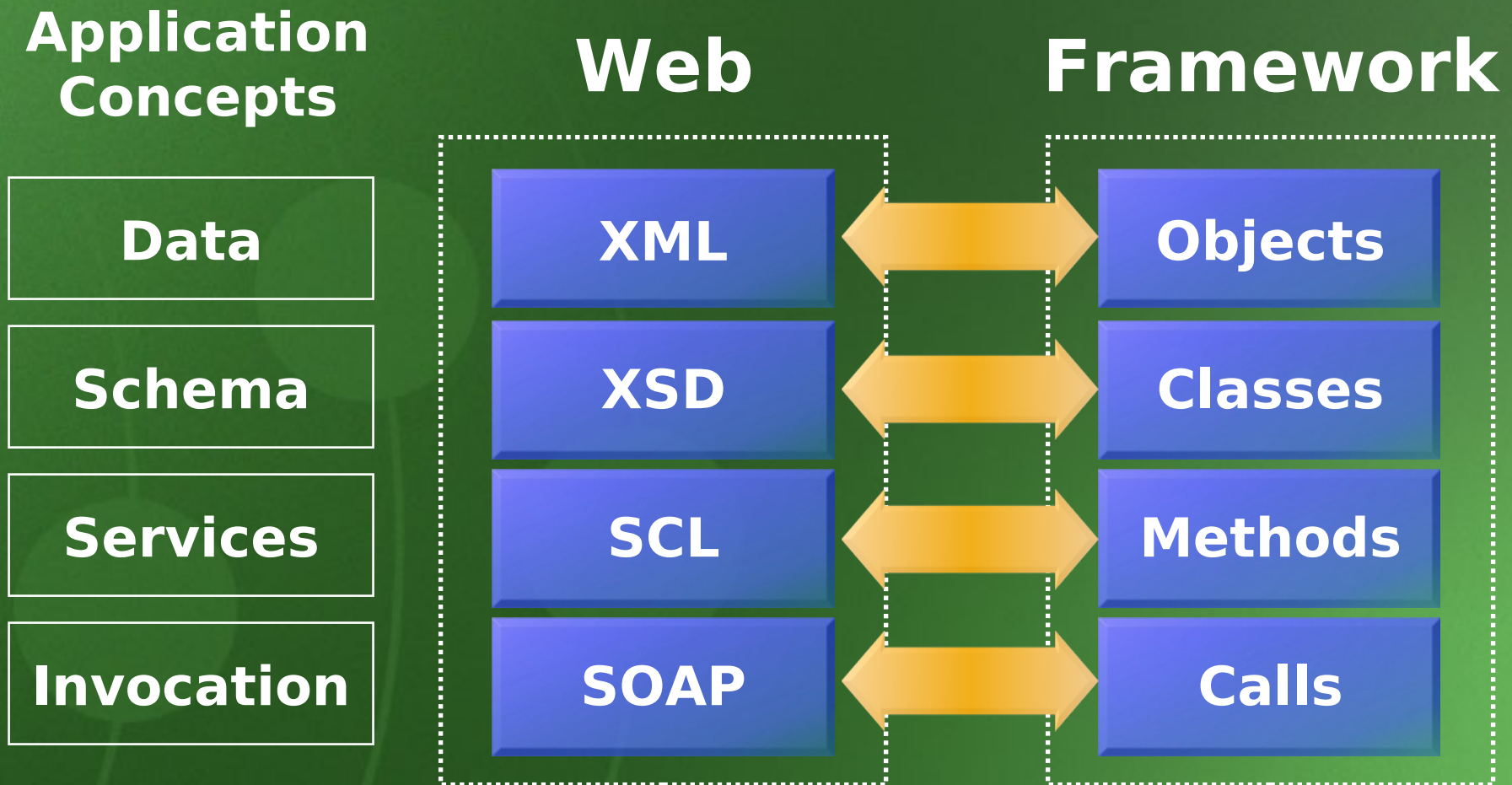
Web

Standards/Practices

- The .NET Framework supports
 - HTML, XML, SOAP, XSLT, XPath...
- Internet Ready distributed apps
 - The traditional tightly connected, stateful model just doesn't work on the Web
 - The .NET Framework favors loosely connected, stateless Web Services
- This affects everything
 - Base services, data access, UI, invocation/activation, programming model

Web Services

The .NET Framework provides a bi-directional mapping



Unify Programming Models

Consistent API availability regardless of language and programming model

.NET Framework

RAD,
Composition,
Delegation

Subclassing,
Power,
Expressiveness

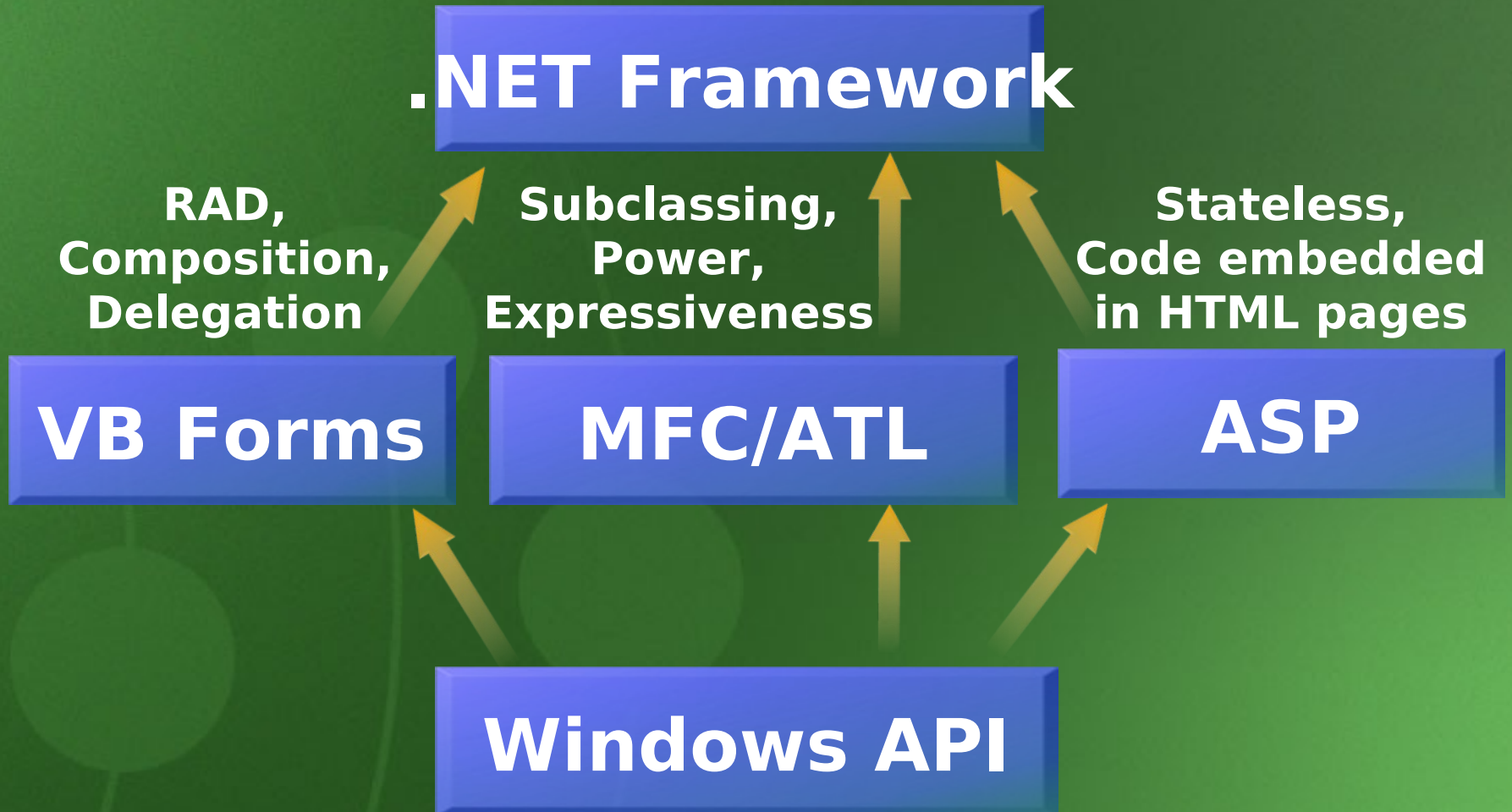
Stateless,
Code embedded
in HTML pages

VB Forms

MFC/ATL

ASP

Windows API



Make It Simple To Use

- **Organization**

- Code organized in hierarchical namespaces and classes

- **Unified type system**

- Everything is an object, no variants, one string type, all character data is Unicode

- **Component Oriented**

- Properties, methods, events, and attributes are first class constructs

- Design time functionality

Factored And Extensible

- The Framework is not a “black box”
- Any .NET class is available for you to extend through inheritance
 - Unlike COM, you’re using and extending the class itself, not a “wrapper”
- Plug-and-play components and subsystems
- Cross-language inheritance!

The .NET Framework

Web Services

User Interface

Data

Xml

Base Class Library

The .NET Framework

System.Web

Services

Description

Discovery

Protocols

Caching

Configuration

UI

HtmlControls

WebControls

Security

SessionState

System.WinForms

Design

ComponentModel

System.Drawing

Drawing2D

Imaging

Printing

Text

System.Data

ADO

Design

SQL

SQLTypes

System.Xml

XSLT

XPath

Serialization

System

Collections

Configuration

Diagnostics

Globalization

IO

Net

Reflection

Resources

Security

ServiceProcess

Text

Threading

Runtime

InteropServices

Remoting

Serialization

Base Framework

System

Collections

Configuration

Diagnostics

Globalization

IO

Net

Reflection

Resources

Security

ServiceProcess

Text

Threading

Runtime

InteropServices

Remoting

Serialization

Data And XML

System.Data

ADO

SQL

Design

SQLTypes

System.Xml

XSLT

Serialization

XPath

Web Forms And

System.Web

Services

Description

Discovery

Protocols

UI

HtmlControls

WebControls

Caching

Security

Configuration

SessionState

Win Forms

System.Windows.Forms

Design

ComponentModel

System.Drawing

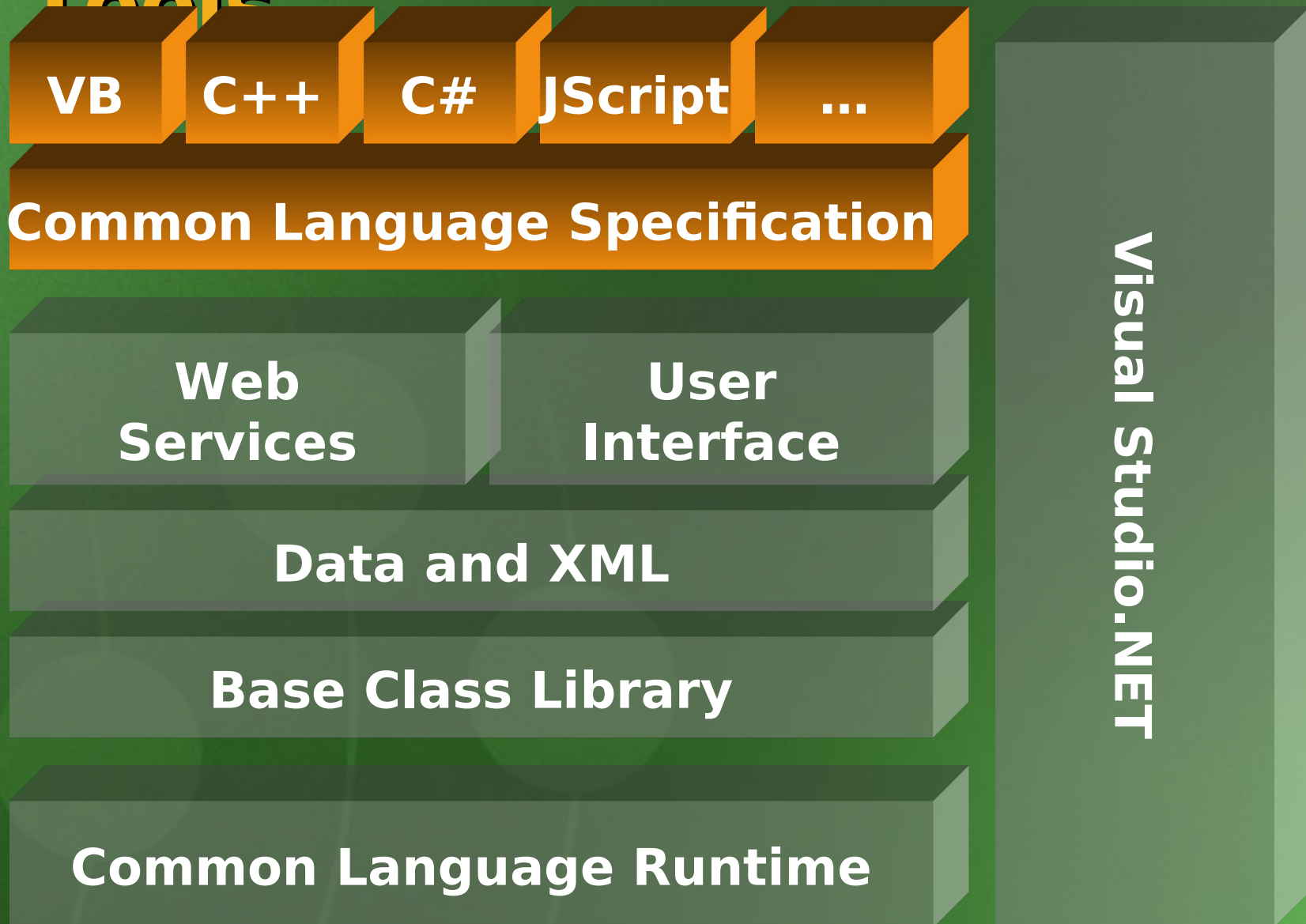
Drawing2D

Printing

Imaging

Text

Framework, Languages, And Tools



Languages

- **The .NET Platform is Language Neutral**
 - All .NET languages are first class players
 - You can leverage your existing skills
- **Common Language Specification**
 - Consumer: Can *use* the .NET Framework
 - Extender: Can *extend* the .NET Framework
- **We are providing**
 - VB, C++, C#, JScript
- **Third-parties are building**
 - API, CORBA, B, Eiffel, Haskell, ML

Visual Basic.NET

- **First class player**
 - VBRUN replaced by .NET Framework
 - Every feature of the .NET Framework is directly available
- **Substantial Language Innovation**
 - Classes, inheritance, constructors, polymorphism, overloading, etc.
 - Structured exceptions
 - One form of assignment

Visual Basic.NET

- **New RAD features**
 - XML designer
 - Component designer for Business Objects, Web Services, etc.
 - Server explorer
 - WebForms designer
- **VBScript becomes full VB**
 - Compiled code in ASP+ pages
- **Leverages your skills**

Managed Extensions To C++

- **Moving your existing code to the .NET Framework**
 - It just works!
- **It's still C++**
 - Minimal extensions, in the ANSI spirit
 - Nothing taken away
- **Power and control**
 - Mixing native and managed code, using non-GC'd data
 - Allows you to gradually migrate
- **Full access to the .NET Framework**

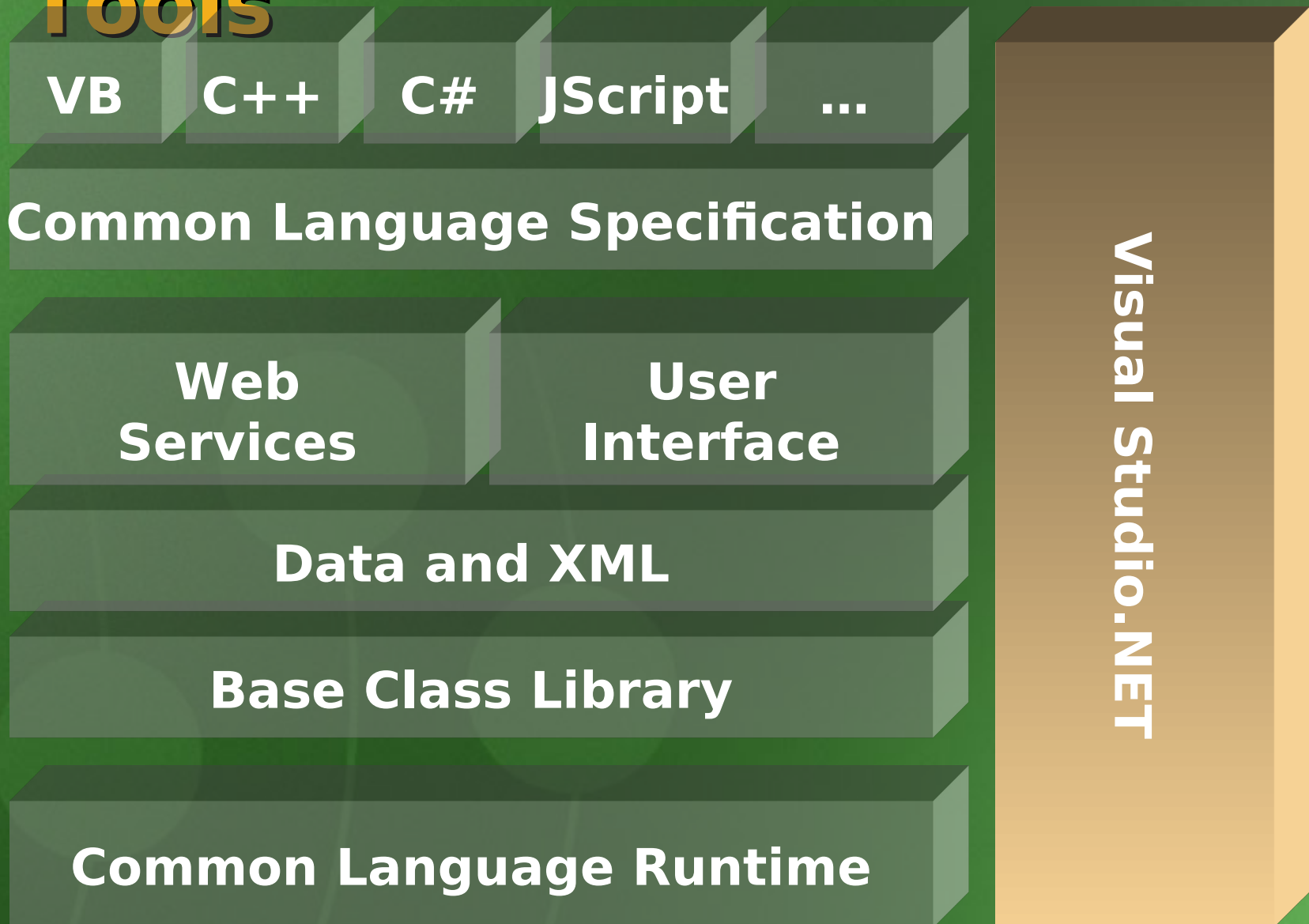
C#

- **The first component oriented language in the C/C++ family**
 - Properties, Methods, Events, Attributes, XML documentation
 - All in one place, no header files, IDL, etc.
 - Can be embedded in ASP+ pages
- **Everything really is an object**
 - Primitive types aren't magic
 - Unified type system == Deep simplicity

C#

- **Enable development of robust and durable software**
 - Designed from the ground up
 - Pervasive versioning considerations in all aspects of language design
- **Preserving your investment**
 - Simplify C++, but preserve heritage
 - Integration through COM and P/Invoke
 - Millions of lines of C# in .NET

Framework, Languages, And Tools



Visual Studio.NET

Dave Mendlen
Lead Product Manager

Visual Studio.NET



Most comprehensive tool for rapidly building enterprise Web applications

- **High productivity tools for team-based enterprise Web development**
- **Most productive tool for creating and consuming Web services**
- **End-to-end tools for enterprise lifecycle productivity**

Demo Architecture

**VISTA.COM
Web Forms**

**Point of Interest
C# Web Service**

**Lodging
VB Web Service**

**Airline
ATL Web Service**

**Products
Web Service**

The background is a solid green color. On the left side, there are three light green circles of varying sizes, connected by thin, curved green lines, creating a decorative pattern.

demo

Visual Studio.NET

Where do **you** want to go today?

Microsoft